

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION**

EXPRESS MOBILE, INC.,

Plaintiff,

v.

ATLASSIAN CORP. PLC and
ATLASSIAN, INC.,

Defendant.

Civil Action No. 6:20-cv-805

JURY TRIAL DEMANDED

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Express Mobile, Inc. (“Express Mobile” or “Plaintiff”), for its complaint against Defendants Atlassian Corp. plc, and Atlassian, Inc. (collectively, “Atlassian” or “Defendants”), alleges the following:

NATURE OF THE ACTION

1. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 1 *et seq.*

THE PARTIES

2. Express Mobile is a corporation organized under the laws of the State of Delaware with a place of business at 38 Washington Street, Novato, CA 94947.

3. Atlassian Corp. plc, is an Australian corporation, and Atlassian, Inc., is a corporation organized under the laws of the State of Delaware, with a place of business at 303 Colorado Street, Suite 1600, Austin, TX 78701. Atlassian, Inc., can be served through its registered agent in Texas, Corporation Service Company, d/b/a CSC-Lawyers Inc., 211 E. 7th Street, Suite 620, Austin, TX 78701.

4. Atlassian offers services throughout the United States, including in this judicial District, and introduces services into the stream of commerce that incorporate infringing technology knowing that they would be used in this judicial District and elsewhere in the United States.

JURISDICTION AND VENUE

5. This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).

6. This Court has personal jurisdiction over Atlassian because it has purposefully availed itself of the rights and benefits of the laws of this State and this District. Atlassian resides in the Western District of Texas by maintaining a regular and established place of business at 303 Colorado Street, Suite 1600, Austin, TX 78701. This Court also has personal jurisdiction over Atlassian because it has done and is doing substantial business in this District, both generally and with respect to the allegations in this complaint, including Atlassian's one or more acts of infringement in this District.

7. Venue is proper in this District under 28 U.S.C. §§ 1391(b) and (c) and 1400(b). Atlassian has committed acts of infringement through provision of its website builder in the Western District of Texas and has at least one regular and established place of business in this District, specifically 303 Colorado Street, Suite 1600, Austin, TX 78701. Atlassian's office in Austin is a physical place in the District, it is an established location where Atlassian's business has been carried out for years, and Atlassian publicly advertises its presence in the District. *See In re Cray, Inc.*, 871 F.3d 1355, 1360-61 (Fed. Cir. 2017).

BACKGROUND

8. Plaintiff Express Mobile is an innovator and leader in the business of developing mobile application and website design and creation platforms. Express Mobile is managed by individuals with many years of technology and business experience. The CEO of Express Mobile,

Steve Rempell, is the inventor of the breakthrough technology held in Express Mobile's patent portfolio. Mr. Rempell has over 50 years' working experience in technology companies, with much of that experience focused on web-based technologies and applications.

9. Before the Express Mobile invention at issue, webpages were created, stored, and rendered using code files that defined all the fixed parameters of the webpage, including, for example, the formatting and location of text, or the location, size, and aspect ratio of images. Typically, webpages could not be viewed during the creation process as they would later appear in the various available browsers or on different devices, and each individual webpage of a website needed to be stored as a separate file. The size and formatting of the stored files led to slow download times to the user's computer, increasing the wait time for a page to load.

10. Express Mobile developed groundbreaking improvements in the process for creating, storing, and building webpages and websites. Express Mobile's invention enables defining the webpage as a collection of user settings, storing information related to those settings in a database, and then later using that information to render a webpage. The page can be viewed, as it is created or edited, in the same manner that it would appear on different types of screens when later accessed. The result is not a collection of computer code, but instead a group of user-selected objects and settings describing the final webpage. These can be saved in a database for ease of access and efficient storage. The invention allows faster loading speeds and permits more efficient storage of the data used to later build the webpages. It also makes changing the webpage more efficient through editing user settings rather than editing multiple lines or versions of code.

11. Defendant Atlassian is a leader in enterprise product and content management. Atlassian offers productivity tools for both small and large businesses. One product, Confluence, is a collaborative information system for large organizations. It allows users without technical training to easily create webpages and other content and share them across the organization. *See*

<https://www.atlassian.com/software/confluence>. Another tool, Trello, is a project management tool that allows enterprise teams to track work items and deadlines by visualizing them as “cards” on a “board.” *See https://trello.com*. Atlassian also offers a project management tool called JIRA, which is widely used by large companies to track issues (such as bugs or feature requests) in software projects. *See https://www.atlassian.com/software/jira*.

12. Confluence allows users to create webpages for people within their organization using menus to control the appearance and content of the final page. Selection of menu items updates the preview of the final page. Trello and JIRA offer the same features – users can share webpages relating to projects with others, and use menus of settings to control the appearance of content, such as cards in Trello and items in JIRA. When users select a setting, those products display the results in real time.

13. Atlassian’s products store information about the selected settings in a database. When a user titles a “To Do” item as “Task 1” in Trello, for example, data regarding that setting, including the chosen title, is sent to Atlassian’s servers, where it is stored in a database for later retrieval. Confluence and JIRA operate similarly.

14. When users access a Trello, Confluence, or JIRA page, the information regarding the settings for the page is retrieved from the database. And JavaScript run time files inspect that retrieved information to generate and display the final webpage according to the settings.

COUNT I – INFRINGEMENT OF U.S. PATENT NO. 6,546,397
(Confluence)

15. The allegations set forth in the foregoing paragraphs 1 through 14 are incorporated into this First Claim for Relief.

16. On April 8, 2003, U.S. Patent No. 6,546,397 (“the ’397 patent”), entitled *Browser Based Web Site Generation Tool and Run Time Engine*, was duly and legally issued by the United

States Patent and Trademark Office. A true and correct copy of the '397 patent is attached as Exhibit A.

17. The claimed invention of the '397 patent resolves technical problems related to website creation and generation. Prior to the invention taught and disclosed in the '397 patent, webpages were generally created, stored, and rendered either by programming directly in HTML, CSS,¹ or JavaScript code, or by using a visual editor that produced HTML files. The result was a collection of pages of computer code – typically HTML, CSS, JavaScript, or Java applets – which defined the visual layout, style, and business logic of websites.

18. Conventional website creation and generation methods suffered from many flaws. Creating a webpage could be cumbersome. Webpages could not be viewed throughout the creation process as they would later appear in various browsers or on different devices. Each individual webpage of a website was stored as a separate HTML, CSS, or JavaScript file, which wasted computer resources and required longer access times in the form of hard drive access while editing websites, and in the form of network traffic while downloading them. Prior-art methods also led to slow downloading of the webpage file to a user's computer and slower rendering by the browser, which increased the wait time for a page to load.

19. Unlike prior-art methods, the '397 patent brings together disparate ideas and concepts for creating, storing, and building webpages. The Express Mobile invention at issue defines webpages as combinations of user-selected objects and settings stored in a database, rather than as combinations of computer code. Because code files do not need to be stored, the page structure – the vast majority of the HTML code itself – is created on the fly each time the page is

¹ CSS, or “Cascading Style Sheets,” is a programming language designed to interoperate with HTML to specify the appearance and placement of web elements.

loaded in a user browser. This unconventional step of building the webpage HTML code on the fly is performed by the run time engine of the invention, using data representative of the user settings. This allows the system to optimize the page based on device-specific information, including the operating system, browser, and screen size. Moreover, the process of defining the webpages is done through a “What You See Is What You Get” or “WYSIWYG” environment, so that, as the page is created or edited, it can be viewed in the same manner it will appear on different types of screens when later accessed.

20. Express Mobile’s patents are directed at a revolutionary technological solution to a technological problem – how to create webpages for the Internet in a manner that permits “What You See Is What You Get” editing, and a number of other improvements over the then-existing methodologies. The claims are not drawn so broadly as to be divorced from the patent-eligible technological improvements described in the specification.

21. The invention claimed in the ’397 patent is not merely the routine or conventional use of website creation systems and methods. Rather, the invention enables the creation of websites through browser-based visual editing tools such as selectable settings panels that describe website elements, with one or more settings corresponding to commands. The invention also enables retrieving that information to generate a website. Those features are implemented exclusively using computer technology, including using virtual machines.

22. The invention claimed in the ’397 patent offers substantial improvements in computer performance and web design. For example, the invention allows for faster loading speeds, more efficient storage of webpage data, and the ability to change the webpage more efficiently by editing user settings rather than multiple versions of code. The invention also permits scaling of webpages and elements within the webpage to most efficiently use the screen

space. Taken separately or together, the claim elements of the invention significantly improve the operation of a computer and the process of web design.

23. The claims of the '397 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '397 patent recite one or more inventive concepts that are rooted in computerized website creation technology, and overcome problems specifically arising in the realm of computerized website creation technologies.

24. The invention claimed in the '397 patent neither preempts all ways of using website or webpage authoring tools nor preempts the use of all website or webpage authoring tools or any other well-known or prior-art technology. Accordingly, each claim of the '397 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

25. Express Mobile is the assignee and owner of the right, title, and interest in and to the '397 patent, including the right to assert all causes of action arising under that patent and the right to any remedies for infringement of it.

26. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringed, either literally or under the doctrine of equivalents, the '397 patent in violation of 35 U.S.C. § 271(a).

27. Upon information and belief, Atlassian has infringed at least claim 1 of the '397 patent.

28. Claim 1 of the '397 patent recites a method to allow users to produce Internet websites on and for computers having a browser and a virtual machine capable of generating displays, said method comprising: (a) presenting a viewable menu having a user-selectable panel of settings describing elements on a website, said panel of settings being presented through a

browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom, and where at least one of said user-selectable settings in said panel corresponds to commands to said virtual machine; (b) generating a display in accordance with one or more user-selected settings substantially contemporaneously with the selection thereof; (c) storing information representative of said one or more user-selected settings in a database; (d) generating a website at least in part by retrieving said information representative of said one or more user-selected settings stored in said database; (e) building one or more webpages to generate said website from at least a portion of said database and at least one run time file, where said at least one run time file uses information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more webpages.

29. Atlassian infringed at least claim 1 of the '397 patent. During the relevant time periods, Confluence practiced a method to allow users to produce Internet webpages on and for computers having a browser and a virtual machine capable of generating displays.

30. By way of example, Confluence presented a viewable menu displaying user-selectable settings that allowed users to specify which elements they wanted to appear on their pages. Additional viewable menus controlled user-selectable settings controlling the appearance of individual web elements on the page. For example, a user could include an image on their page and select how big they wanted the image to be.

31. By way of further example, Confluence presented a viewable menu displaying user-selectable settings that allowed users to specify the design of the pages, including settings that corresponded to subsections of their webpage, as well as the font size, highlights, and effects of the text on the pages. By way of further example, Confluence allowed users to click and move elements appearing on the pages, placing them in their desired locations.

32. Once a user selected any of these settings, Confluence updated the pages in accordance with the selected settings substantially contemporaneously with the selection thereof.

33. On information and belief, Confluence stored user-selected settings in a database.

34. Those user-selectable settings corresponded to commands to a virtual machine.

When a setting was selected, Confluence used it to generate JSON code. One or more run time files containing HTML and JavaScript code communicated with the Confluence web server to send and retrieve the encoded user-selected settings and thus generate the page.

35. By way of example, when a user first loaded Confluence, the Confluence web server sent the run time files to the user's web browser. The run time files then communicated with the Confluence web server to retrieve the user-selectable settings stored in the database, and used them to generate virtual machine commands in the form of JSON code, which was sent to the web browser. The web browser's virtual machine executed the JSON code. That execution, in combination with the commands in the run time files, generated the page in accordance with the saved settings.

36. Atlassian was made aware of the '397 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '397 patent.

37. Upon information and belief, since at least the time Atlassian received notice, Atlassian induced others to infringe at least claim 1 of the '397 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Confluence constituted direct infringement of at least one claim of the '397 patent. In particular, Atlassian's actions that aided and abetted others such as customers, clients, partners, developers, and end users to infringe included advertising Confluence as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian

engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian had actual knowledge of the '397 patent and knowledge that its acts were inducing infringement of the '397 patent since at least the date Atlassian received notice that such activities infringed the '397 patent.

38. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Confluence. For example, Atlassian ran an advertisement explaining that Confluence "comes equipped with best-practice templates," allowing users to easily create webpages.

39. Since February 6, 2020, Atlassian's infringement of the '397 patent has been willful.

40. Atlassian's infringement damaged and injured Express Mobile.

COUNT II – INFRINGEMENT OF U.S. PATENT NO. 6,546,397
(Trello)

41. The allegations set forth in the foregoing paragraphs 1 through 40 are incorporated into this Second Claim for Relief.

42. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringed, either literally or under the doctrine of equivalents, one or more claims of the '397 patent in violation of 35 U.S.C. § 271(a).

43. Upon information and belief, Atlassian infringed at least claim 1 of the '397 patent. During the relevant time periods, Trello practiced a method to allow users to produce Internet webpages on and for computers having a browser and a virtual machine capable of generating displays.

44. By way of example, Trello presented a viewable menu displaying user-selectable settings that allowed users to specify which elements they wanted to appear on their pages. Those

elements on the webpage were organized into columns of “cards” containing text, images, and icons. Additional viewable menus controlled user-selectable settings controlling the appearance of individual elements on the cards. For example, a user could add images and icons to selected cards.

45. By way of further example, Trello presented a viewable menu displaying user-selectable settings that allowed users to specify the design of the pages, including settings that corresponded to subsections of the cards, as well as the font size, highlights, and effects of the text on the cards. By way of further example, Trello allowed users to click and move cards on the pages, placing them in different locations.

46. Once a user selected any of these settings, Trello updated the pages in accordance with the selected settings substantially contemporaneously with the selection thereof.

47. On information and belief, Trello stored user-selected settings in a database.

48. Those user-selectable settings corresponded to commands to a virtual machine. When a setting was selected, Trello used it to generate JSON code. One or more run time files containing HTML and JavaScript code communicated with the Trello web server to send and retrieve the encoded user-selected settings and thus generate the page.

49. By way of example, when a user first loaded Trello, the Trello web server sent the run time files to the user’s web browser. The run time files then communicated with the Trello web server to retrieve the user-selectable settings stored in the database, and used them to generate virtual machine commands in the form of JSON code, which was sent to the web browser. The web browser’s virtual machine executed the JSON code. That execution, in combination with the commands in the run time files, generated the page in accordance with the saved settings.

50. Atlassian was made aware of the ’397 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the ’397 patent.

51. Upon information and belief, since at least the time Atlassian received notice, Atlassian induced others to infringe at least claim 1 of the '397 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Trello constituted direct infringement of at least one claim of the '397 patent. In particular, Atlassian's actions that aided and abetted others such as customers, clients, partners, developers, and end users to infringe included advertising Trello as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian had actual knowledge of the '397 patent and knowledge that its acts were inducing infringement of the '397 patent since at least the date Atlassian received notice that such activities infringed the '397 patent.

52. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Trello. For example, Atlassian ran an advertisement showing that users can collaborate with others to organize cards "in a fun, flexible, and rewarding way."

53. Another advertisement showed Trello's visual interface for creating, editing, and moving cards in a visual manner.

54. Since February 6, 2020, Atlassian's infringement of the '397 patent has been willful.

55. Atlassian's infringement damaged and injured Express Mobile.

COUNT III – INFRINGEMENT OF U.S. PATENT NO. 6,546,397
(JIRA)

56. The allegations set forth in the foregoing paragraphs 1 through 55 are incorporated into this Third Claim for Relief.

57. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringed, either literally or under the doctrine of equivalents, one or more claims of the '397 patent in violation of 35 U.S.C. § 271(a).

58. Upon information and belief, Atlassian infringed at least claim 1 of the '397 patent. During the relevant time periods, JIRA practiced a method to allow users to produce Internet webpages, those pages representing tracked bug reports, tasks, or issues, on and for computers having a browser and a virtual machine capable of generating displays.

59. By way of example, JIRA presented a viewable menu displaying user-selectable settings that allowed users to specify which elements they wanted to appear on their pages. Additional viewable menus controlled user-selectable settings controlling the appearance of individual elements on the issue webpage. For example, a user could include screenshots and other images on the page.

60. By way of further example, JIRA presented a viewable menu displaying user-selectable settings that allowed users to specify the design of the pages, including settings that corresponded to various fields of an issue, as well as the font size and other formatting of the text content in an issue.

61. Once a user selected any of these settings, JIRA updated the pages in accordance with the selected settings substantially contemporaneously with the selection thereof.

62. On information and belief, JIRA stored user-selected settings in a database.

63. Those user-selectable settings corresponded to commands to a virtual machine. When a setting was selected, JIRA used it to generate JSON code. One or more run time files containing HTML and JavaScript code communicated with the JIRA web server to send and retrieve the encoded user-selected settings and thus generate the page.

64. By way of example, when a user first loaded JIRA, the JIRA web server sent the run time files to the user's web browser. The run time files then communicated with the JIRA web server to retrieve the user-selectable settings stored in the database, and used them to generate virtual machine commands in the form of JSON code, which was sent to the web browser. The web browser's virtual machine executed the JSON code. That execution, in combination with the commands in the run time files, generated the page in accordance with the saved settings.

65. Atlassian was made aware of the '397 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '397 patent.

66. Upon information and belief, since at least the time Atlassian received notice, Atlassian induced others to infringe at least claim 1 of the '397 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of JIRA constituted direct infringement of at least one claim of the '397 patent. In particular, Atlassian's actions that aided and abetted others such as customers, clients, partners, developers, and end users to infringe included advertising JIRA as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian had actual knowledge of the '397 patent and knowledge that its acts were inducing infringement of the '397 patent since at least the date Atlassian received notice that such activities infringed the '397 patent.

67. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of JIRA. For example, Atlassian ran an advertisement showing that users can create a webpage summarizing data entered into the platform "all in a few clicks."

68. Other JIRA advertisements touted JIRA's interface allowing for the easy creation of "flexible" and "customizable" pages.

69. Since February 6, 2020, Atlassian's infringement of the '397 patent has been willful.

70. Atlassian's infringement damaged and injured Express Mobile.

COUNT IV – INFRINGEMENT OF U.S. PATENT NO. 9,063,755
(Confluence)

71. The allegations set forth in the foregoing paragraphs 1 through 70 are incorporated into this Fourth Claim for Relief.

72. On June 23, 2015, U.S. Patent No. 9,063,755 ("the '755 patent"), entitled *Systems and Methods for Presenting Information on Mobile Devices*, was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '755 patent is attached as Exhibit B.

73. The invention claimed in the '755 patent resolves technical problems related to generating and distributing dynamic content on a device display, such as the display of a mobile device. Before the patents-in-suit, content and applications for device displays were generally created using code written for each individual type of device. As device types proliferated, programming content and applications for each device became increasingly expensive and time-consuming. Doing so also limited the ability of providers to update the capabilities of, and increase the available content for, many devices.

74. The invention of the '755 patent resolves technical problems related to generating and distributing dynamic content on a device display. The invention features a computer memory and an authoring tool or Player configured to define a User Interface ("UI") object for display on the device, where the defined UI object corresponds to a web component and where each UI object

is either: (1) selected by a user or (2) automatically selected by the system as a preferred UI object corresponding to a symbolic name of the web component. Additionally, the computer memory and the authoring tool or Player are configured to build an Application consisting of one or more webpage views to provide for the display of at least a portion of one or more of the webpages. These features are exclusively implemented using computer technology.

75. Unlike methods in the prior art, the '755 patent brings together disparate ideas and concepts for generating and distributing content suitable for display on different devices with varying characteristics using a combination of device-independent and device- and platform-dependent code. This can include building a webpage or application using a "What You See Is What You Get" or "WYSIWYG" environment, so that, as the page or app is created or edited, it can be viewed in the same manner it will appear on different types of screens when later accessed. The invention can also include an authoring tool that can create an Application, where the Application is device-independent code, and a Player, where the Player is device-dependent code. The Player enables the Application to function on a variety of devices or platforms, with differing functionality. This enables users of the authoring tool to create and distribute device-independent Applications for different device types, without individually tailoring the device-independent Applications for each device type.

76. The claims of the '755 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '755 patent recite one or more inventive concepts that are rooted in the computerized generation of content on a device display, such as a mobile device, and overcome problems specifically arising in the realm of computerized display content generation technologies.

77. The claims of the '755 patent recite an invention that is not merely the routine or conventional use of systems and methods for the computerized generation of content on a device display. Instead, the invention describes systems for use with devices with authoring tools or Players specific to each device and Applications that are independent of the device.

78. The invention claimed in the '755 patent offers substantial improvements in device performance and web or application design. For example, the invention allows for faster loading speeds, more efficient storage of webpage or application data, and the ability to change a webpage or application more efficiently by editing user settings rather than multiple versions of code. The invention also permits scaling of webpages and elements within the webpage, or applications and elements within the application, to most efficiently use the screen space. Taken separately or together, the claim elements of the invention significantly improve the operation of a computer and the process of web design.

79. The invention claimed in the '755 patent neither preempts all ways for the computerized generation of content on a device display, such as a mobile device, nor preempts the use of all authoring tools or Players for the computerized generation of content on a device display, such as a mobile device, or any other well-known or prior-art technology.

80. Accordingly, each claim of the '755 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

81. Express Mobile is the assignee and owner of the right, title, and interest in and to the '755 patent, including the right to assert all causes of action arising under the patent and the right to any remedies for infringement of it.

82. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '755 patent in violation of 35 U.S.C. § 271(a).

83. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 23 of the '755 patent.

84. Claim 23 of the '755 patent recites a method of providing information to a device having a display from a web component of a web service to a device on a network, said method comprising: accepting, on the device, a first code over the network, where said first code is device-dependent; accepting, on the device, a second code over the network, where said second code is device-independent and includes a plurality of symbolic names of inputs and outputs associated with the web service; and executing said first code on the device, where the symbolic names are provided from a registry of one or more web components related to inputs and outputs of a web service obtainable over a network, where the web service requires both an input symbolic name and one or more associated input values and returns one or more output values having an associated output symbolic name, and where the registry includes (a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, and (b) the address of the web service; where said executing includes: processing said symbolic names of the second code on the device, transmitting processed instructions from the device to the web service, and accepting a third code on the device over the network, where said third code is a device-independent third code including the output of the web component provided by the web service over the network and in response to the second code.

85. Atlassian infringes claim 23 of the '755 patent through a combination of features that collectively practice each limitation of claim 23. Confluence practices a method for providing information to a device having a display, including a web browser. The information comes from a web service, the Confluence web server.

86. Confluence accepts and executes device- and platform-dependent code from the Confluence web server, including HTML, CSS, and JavaScript. It also accepts device-independent code from the Confluence web server, which includes symbolic names of inputs and outputs. Unlike the device-independent code, the device- and platform-dependent code is written for specific device platforms and devices, such as browsers, laptops, tablets, or smartphones.

87. The symbolic names are provided from a registry of web components related to inputs and outputs obtainable over a network. The web components include images, text blocks, tables, and other common web components.

88. The registry that Confluence uses contains the address of a web service available over a network (the Confluence web server) and symbolic names related to inputs and outputs of the web service. The symbolic names are character strings that do not contain either a persistent address or pointer to an output value. The Confluence web server accepts both an input symbolic name and one or more associated input values from a user and returns one or more outputs having an associated symbolic name.

89. When the browser executes the code provided to it, it processes the symbolic names and transmits instructions back to the web service. In response, it accepts new, device-independent code from the web service.

90. Atlassian was made aware of the '755 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '755 patent.

91. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 23 of the '755 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Confluence constitutes direct infringement of at least one claim of the '755 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Confluence as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '755 patent and knowledge that its acts were inducing infringement of the '755 patent since at least the date Atlassian received notice that such activities infringed the '755 patent.

92. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Confluence. For example, Atlassian ran an advertisement explaining that Confluence "comes equipped with best-practice templates," allowing users to easily create webpages.

93. Since February 6, 2020, Atlassian's infringement of the '755 patent has been willful.

94. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT V – INFRINGEMENT OF U.S. PATENT NO. 9,063,755
(Trello)

95. The allegations set forth in the foregoing paragraphs 1 through 94 are incorporated into this Fifth Claim for Relief.

96. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '755 patent in violation of 35 U.S.C. § 271(a).

97. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 23 of the '755 patent. Trello practices a method for providing information to a device having a display, including a web browser. The information comes from a web service, the Trello web server. The webpages comprise columns of cards containing text, images, and icons.

98. Trello accepts and executes device- and platform-dependent code from the Trello web server, including HTML, CSS, and JavaScript. It also accepts device-independent code from the Trello web server, which includes symbolic names of inputs and outputs. Unlike the device-independent code, the device- and platform-dependent code is written for specific device platforms and devices, such as browsers, laptops, tablets, or smartphones.

99. The symbolic names are provided from a registry of web components related to inputs and outputs obtainable over a network. The web components include images, text blocks, cards, and other common web components.

100. The registry that Trello uses contains the address of a web service available over a network (the Trello web server) and symbolic names related to inputs and outputs of the web service. The symbolic names are character strings that do not contain either a persistent address or pointer to an output value. The Trello web server accepts both an input symbolic name and one or more associated input values from a user and returns one or more outputs having an associated symbolic name.

101. When the browser executes the code provided to it, it processes the symbolic names and transmits instructions back to the web service. In response, it accepts new, device-independent code from the web service.

102. Atlassian was made aware of the '755 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '755 patent.

103. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 23 of the '755 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Trello constitutes direct infringement of at least one claim of the '755 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Trello as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '755 patent and knowledge that its acts were inducing infringement of the '755 patent since at least the date Atlassian received notice that such activities infringed the '755 patent.

104. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Trello. For example, Atlassian ran an advertisement showing that users can collaborate with others to organize cards “in a fun, flexible, and rewarding way.”

105. Another advertisement showed Trello's visual interface for creating, editing, and moving cards in a visual manner.

106. Since February 6, 2020, Atlassian's infringement of the '755 patent has been willful.

107. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT VI – INFRINGEMENT OF U.S. PATENT NO. 9,063,755
(JIRA)

108. The allegations set forth in the foregoing paragraphs 1 through 107 are incorporated into this Sixth Claim for Relief.

109. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '755 patent in violation of 35 U.S.C. § 271(a).

110. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 23 of the '755 patent. JIRA practices a method for providing information to a device having a display, including a web browser. The information comes from a web service, the JIRA web server.

111. JIRA accepts and executes device- and platform-dependent code from the JIRA web server, including HTML, CSS, and JavaScript. It also accepts device-independent code from the JIRA web server, which includes symbolic names of inputs and outputs. Unlike the device-independent code, the device- and platform-dependent code is written for specific device platforms and devices, such as browsers, laptops, tablets, or smartphones.

112. The symbolic names are provided from a registry of web components related to inputs and outputs obtainable over a network. The web components include images, text blocks, tables, and other common web components.

113. The registry that JIRA uses contains the address of a web service available over a network (the JIRA web server) and symbolic names related to inputs and outputs of the web service. The symbolic names are character strings that do not contain either a persistent address or pointer to an output value. The JIRA web server accepts both an input symbolic name and one

or more associated input values from a user and returns one or more outputs having an associated symbolic name.

114. When the browser executes the code provided to it, it processes the symbolic names and transmits instructions back to the web service. In response, it accepts new, device-independent code from the web service.

115. Atlassian was made aware of the '755 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '755 patent.

116. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 23 of the '755 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of JIRA constitutes direct infringement of at least one claim of the '755 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising JIRA as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '755 patent and knowledge that its acts were inducing infringement of the '755 patent since at least the date Atlassian received notice that such activities infringed the '755 patent.

117. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of JIRA. For example, Atlassian ran an advertisement showing that users can create a webpage summarizing data entered into the platform "all in a few clicks."

118. Other JIRA advertisements touted JIRA's interface allowing for the easy creation of "flexible" and "customizable" pages.

119. Since February 6, 2020, Atlassian's infringement of the '755 patent has been willful.

120. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT VII – INFRINGEMENT OF U.S. PATENT NO. 9,471,287
(Confluence)

121. The allegations set forth in the foregoing paragraphs 1 through 120 are incorporated into this Seventh Claim for Relief.

122. On October 18, 2016, U.S. Patent No. 9,471,287 ("the '287 patent"), entitled *Systems and Methods for Integrating Widgets on Mobile Devices*, was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '287 patent is attached as Exhibit C.

123. The invention claimed in the '287 patent resolves technical problems related to generating and distributing dynamic content on a device display, such as the display of a mobile device. Before the patents-in-suit, content and applications for device displays were generally created using code written for each individual type of device. As device types proliferated, programming content and applications for each device became increasingly expensive and time-consuming. Doing so also limited the ability of providers to update the capabilities of, and increase the available content for, many devices.

124. The invention of the '287 patent resolves technical problems related to generating and distributing dynamic content on a device display, such as the display of a mobile device. The invention of the '287 patent features a registry and an authoring tool or Player configured to define a User Interface ("UI") object for display on the device, where the UI object corresponds to a web component. Each UI object is either: (1) selected by a user or (2) automatically selected by the

system as a preferred UI object corresponding to a symbolic name of the web component and used to produce an Application, where the Application is a device-independent code, and a Player, where the Player is a device- and platform-dependent code. The Application and Player (1) enable the device to provide one or more input values and corresponding input symbolic name to the web service and (2) enable the web service to use the input symbolic name and one or more user-provided input values to generate one or more output values having an associated output symbolic name, while (3) the Player receives the output symbolic name and one or more corresponding output values and provides instructions for the display of the device to present an output value in the defined UI object. These features are exclusively implemented using computer technology.

125. Unlike methods in the prior art, the '287 patent brings together disparate ideas and concepts for generating and distributing content on different device displays. This can include building a webpage or application using a "What You See Is What You Get" or "WYSIWYG" environment, so that, as the page or app is created or edited, it can be viewed in the same manner it will appear on different types of screens when later accessed. The invention can also include an authoring tool that can create an Application, where the Application is device-independent code, and a Player, where the Player is device- and platform-dependent code. The Player enables the Application to function on a variety of devices or platforms, with differing functionality. This enables users of the authoring tool to create and distribute device-independent Applications for different device types, without individually tailoring the device-independent Applications for each device type.

126. The claims of the '287 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '287 patent recite one or more inventive concepts that are rooted in the computerized generation of content on a device display, such as a mobile device, and

overcome problems specifically arising in the realm of computerized display content generation technologies.

127. The claims of the '287 patent recite an invention that is not merely the routine or conventional use of systems and methods for the computerized generation of content on a device display. Instead, the invention features systems that can be used with devices and methods of using the systems with authoring tools or Players specific to each device platform and Applications that are independent of the device.

128. The invention claimed in the '287 patent offers substantial improvements in device performance and web or application design. For example, the invention allows for faster loading speeds, more efficient storage of webpage or application data, and the ability to change a webpage or application more efficiently by editing user settings rather than multiple versions of code. The invention also permits scaling of webpages and elements within the webpage, or applications and elements within the application, to most efficiently use the screen space. Taken separately or together, the claim elements of the invention significantly improve the operation of a computer and the process of web design.

129. The invention claimed in the '287 patent neither preempts all ways for the computerized generation of content on a device display, such as a mobile device, nor preempts the use of all authoring tools or Players for the computerized generation of content on a device display, such as a mobile device, or any other well-known or prior-art technology.

130. Accordingly, each claim of the '287 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

131. Express Mobile is the assignee and owner of the right, title, and interest in and to the '287 patent, including the right to assert all causes of action arising under the patent and the right to any remedies for infringement of it.

132. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '287 patent in violation of 35 U.S.C. § 271(a).

133. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the '287 patent.

134. Claim 15 of the '287 patent recites a method of displaying content on a display of a device having a Player, where said Player is a device- and platform-dependent code, said method comprising: defining a user interface (“UI”) object for presentation on the display, where said UI object corresponds to a web component included in a registry of one or more web components selected from a group consisting of an input of a web service and an output of the web service where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and where the registry includes: (a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of the web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, and (b) an address of the web service, and where each defined UI object is either: (1) selected by a user of an authoring tool; or (2) automatically selected by a system as a preferred UI object corresponding to a symbolic name of the web component selected by the user of the authoring tool; selecting the symbolic name from said web component corresponding to the defined UI object, where the selected symbolic name has an associated data format class type corresponding to a subclass of UI objects that support the data format type of the symbolic name, and has the

preferred UI object; associating the selected symbolic name with the defined UI object; and producing an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, wherein, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, (1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, (2) the web service uses the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, (3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.

135. Atlassian infringes claim 15 of the '287 patent through a combination of features that collectively practice each limitation of claim 15. Confluence practices a method for displaying content on a display of a computer device having a Player – HTML, CSS, and JavaScript code written for a particular device platform, such as browsers, laptops, tablets, and smartphones. The content is provided to the user through a web browser on webpages.

136. The method includes computer memory that stores a registry of symbolic names associated with web components – including images, text blocks, tables, and other web components. These components are related to inputs and outputs of a web service – the Confluence web server – obtainable over a network, such as the Internet or corporate intranet.

137. The names stored in the registry are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects.

Confluence's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

138. The Confluence registry also includes the address of the Confluence web server, a web service.

139. Confluence also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

140. The authoring tool then produces an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player and Application Code collaborate to provide the page to the user.

141. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

142. Atlassian was made aware of the '287 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '287 patent.

143. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '287 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Confluence constitutes direct infringement of at least one claim of the '287 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Confluence as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '287 patent and knowledge that its acts were inducing infringement of the '287 patent since at least the date Atlassian received notice that such activities infringed the '287 patent.

144. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Confluence. For example, Atlassian ran an advertisement explaining that Confluence "comes equipped with best-practice templates," allowing users to easily create webpages.

145. Since February 6, 2020, Atlassian's infringement of the '287 patent has been willful.

146. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT VIII – INFRINGEMENT OF U.S. PATENT NO. 9,471,287
(Trello)

147. The allegations set forth in the foregoing paragraphs 1 through 146 are incorporated into this Eighth Claim for Relief.

148. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '287 patent in violation of 35 U.S.C. § 271(a).

149. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the '287 patent. Trello practices a method for displaying dynamic content on a display of a computer device having a Player – HTML, CSS, and JavaScript code written for a particular device platform, such as browsers, laptops, tablets, and smartphones. The content is provided to the user through a web browser on webpages.

150. The method includes computer memory that stores a registry of symbolic names associated with web components – including images, text blocks, tables, and other web components. These components are related to inputs and outputs of a web service – the Trello web server – obtainable over a network, such as the Internet.

151. The names stored in the registry are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects. Trello's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

152. The Trello registry also includes the address of the Trello web server, a web service.

153. Trello also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web

components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

154. The authoring tool then produces an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player and Application Code collaborate to provide the page to the user.

155. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

156. Atlassian was made aware of the '287 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '287 patent.

157. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '287 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Trello constitutes direct infringement of at least

one claim of the '287 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Trello as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '287 patent and knowledge that its acts were inducing infringement of the '287 patent since at least the date Atlassian received notice that such activities infringed the '287 patent.

158. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Trello. For example, Atlassian ran an advertisement showing that users can collaborate with others to organize cards "in a fun, flexible, and rewarding way."

159. Another advertisement showed Trello's visual interface for creating, editing, and moving cards in a visual manner.

160. Since February 6, 2020, Atlassian's infringement of the '287 patent has been willful.

161. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT IX – INFRINGEMENT OF U.S. PATENT NO. 9,471,287
(JIRA)

162. The allegations set forth in the foregoing paragraphs 1 through 161 are incorporated into this Ninth Claim for Relief.

163. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '287 patent in violation of 35 U.S.C. § 271(a).

164. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the '287 patent. JIRA practices a method for displaying content on a display of a computer device having a Player – HTML, CSS, and JavaScript code written for a particular device, such as laptops, tablets, and smartphones. The content is provided to the user through a web browser on webpages.

165. The method includes computer memory that stores a registry of symbolic names associated with web components – including images, issue fields, and other web components. These components are related to inputs and outputs of a web service – the JIRA web server – obtainable over a network, such as the Internet or corporate intranet.

166. The names stored in the registry are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects. JIRA's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

167. The JIRA registry also includes the address of the JIRA web server, a web service.

168. JIRA also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user

of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

169. The authoring tool then produces an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player and Application Code collaborate to provide the page to the user.

170. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

171. Atlassian was made aware of the '287 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '287 patent.

172. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '287 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of JIRA constitutes direct infringement of at least one claim of the '287 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising JIRA as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '287

patent and knowledge that its acts were inducing infringement of the '287 patent since at least the date Atlassian received notice that such activities infringed the '287 patent.

173. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of JIRA. For example, Atlassian ran an advertisement showing that users can create a webpage summarizing data entered into the platform “all in a few clicks.”

174. Other JIRA advertisements touted JIRA’s interface allowing for the easy creation of “flexible” and “customizable” pages.

175. Since February 6, 2020, Atlassian’s infringement of the '287 patent has been willful.

176. Atlassian’s infringement has damaged and continues to damage and injure Express Mobile.

COUNT X – INFRINGEMENT OF U.S. PATENT NO. 9,928,044
(Confluence)

177. The allegations set forth in the foregoing paragraphs 1 through 176 are incorporated into this Tenth Claim for Relief.

178. On March 27, 2018, U.S. Patent No. 9,928,044 (“the '044 patent”), entitled *Systems and Methods for Programming Mobile Devices*, was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '044 patent is attached as Exhibit D.

179. The invention claimed in the '044 patent resolves technical problems related to generating and distributing dynamic content on a device display, such as the display of a mobile device. Before the patents-in-suit, content and applications for device displays were generally created using code written for each individual type of device. As device types proliferated, programming content and applications for each device became increasingly expensive and time-

consuming. Doing so also limited the ability of providers to update the capabilities of, and increase the available content for, many devices.

180. The invention of the '044 patent resolves technical problems related to generating and distributing dynamic content on a device display, such as the display of a mobile device. The invention features a computer memory and an authoring tool or Player configured to define a User Interface ("UI") object for display on the device, where the defined UI object corresponds to a web component and where each UI object is either: (1) selected by a user or (2) automatically selected by the system as a preferred UI object corresponding to a symbolic name of the web component. Additionally, the computer memory and the authoring tool or Player are configured to build an Application consisting of one or more webpage views to provide for the display of at least a portion of one or more of the webpages. These features are exclusively implemented using computer technology.

181. Unlike methods in the prior art, the '044 patent brings together disparate ideas and concepts for generating and distributing content on different device displays. This can include building a webpage or application using a "What You See Is What You Get" or "WYSIWYG" environment, so that, as the page or app is created or edited, it can be viewed in the same manner it will appear on different types of screens when later accessed. The invention can also include an authoring tool that can create an Application, where the Application is device-independent code, and a Player, where the Player is device- and platform-dependent code. The Player enables the Application to function on a variety of devices or platforms, with differing functionality. This enables users of the authoring tool to create and distribute device-independent Applications for different device types, without individually tailoring the device-independent Applications for each device type.

182. The claims of the '044 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '044 patent recite one or more inventive concepts that are rooted in the computerized generation of content on a device display, such as a mobile device, and overcome problems specifically arising in the realm of computerized display content generation technologies.

183. The claims of the '044 patent recite an invention that is not merely the routine or conventional use of systems and methods for the computerized generation of content on a device display. Instead, the invention describes systems for use with devices with authoring tools or Players specific to each device and Applications that are independent of the device.

184. The invention claimed in the '044 patent offers substantial improvements in device performance and web or application design. For example, the invention allows for faster loading speeds, more efficient storage of webpage or application data, and the ability to change a webpage or application more efficiently by editing user settings rather than multiple versions of code. The invention also permits scaling of webpages and elements within the webpage, or applications and elements within the application, to most efficiently use the screen space. Taken separately or together, the claim elements of the invention significantly improve the operation of a computer and the process of web design.

185. The invention claimed in the '044 patent neither preempts all ways for the computerized generation of content on a device display, such as a mobile device, nor preempts the use of all authoring tools or Players for the computerized generation of content on a device display, such as a mobile device, or any other well-known or prior-art technology.

186. Accordingly, each claim of the '044 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

187. Express Mobile is the assignee and owner of the right, title, and interest in and to the '044 patent, including the right to assert all causes of action arising under the patent and the right to any remedies for infringement of it.

188. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '044 patent in violation of 35 U.S.C. § 271(a).

189. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the '044 patent.

190. Claim 15 of the '044 patent recites a method of displaying content on a display of a device having a Player and non-volatile computer memory storing symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (“UI”) objects that support the data format type of the symbolic name, and where each symbolic name has a preferred UI object, and an address of the web service, said method comprising: defining a UI object for presentation on the display, where said UI object corresponds to a web component included in the computer memory, where said web component is selected from a group consisting of an input of a web service and an output of the web service where each defined UI object is either: (1) selected by a user of the authoring tool; or (2) automatically selected by the system as the preferred UI object corresponding to a symbolic name of the web component

selected by the user of the authoring tool; and selecting the symbolic name from said web component corresponding to the defined UI object, associating the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name; storing information representative of said defined UI object and related settings in a database; retrieving said information representative of said one or more UI object settings stored in said database; and building an Application consisting of one or more webpage views from at least a portion of said database using the Player, where said Player uses information stored in said database to generate for the display of at least a portion of said one or more webpages, wherein, when the Application and Player are provided to the device and executed on the device, and when the user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, (1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, (2) the web service uses the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, (3) the Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.

191. Atlassian infringes claim 15 of the '044 patent through a combination of features that collectively practice each limitation of claim 15. Confluence practices a method for displaying content on a display of a computer device. The content is provided to the user through a web browser on webpages.

192. The method includes non-volatile computer memory that stores symbolic names associated with web components – including images, text blocks, tables, and other web

components. These components are related to inputs and outputs of a web service – the Confluence web server – obtainable over a network, such as the Internet or a corporate intranet.

193. The names stored in the computer memory are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects. Confluence's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

194. The Confluence computer memory also includes the address of the Confluence web server, a web service.

195. Confluence also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

196. The authoring tool then stores the symbolic code with the associated settings for the user-defined object in a database. It retrieves that information to produce an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-

dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player code and Application code collaborate to provide the page to the user.

197. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

198. Atlassian was made aware of the '044 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '044 patent.

199. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '044 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Confluence constitutes direct infringement of at least one claim of the '044 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Confluence as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '044 patent and knowledge that its acts were inducing infringement of the '044 patent since at least the date Atlassian received notice that such activities infringed the '044 patent.

200. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Confluence. For example, Atlassian ran an advertisement explaining that

Confluence “comes equipped with best-practice templates,” allowing users to easily create webpages.

201. Since February 6, 2020, Atlassian’s infringement of the ’044 patent has been willful.

202. Atlassian’s infringement has damaged and continues to damage and injure Express Mobile.

COUNT XI – INFRINGEMENT OF U.S. PATENT NO. 9,928,044
(Trello)

203. The allegations set forth in the foregoing paragraphs 1 through 202 are incorporated into this Eleventh Claim for Relief.

204. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the ’044 patent in violation of 35 U.S.C. § 271(a).

205. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the ’044 patent. Trello practices a method for displaying content on a display of a computer device. The content is provided to the user through a web browser on webpages.

206. The method includes non-volatile computer memory that stores symbolic names associated with web components – including images, text blocks, cards, and other web components. These components are related to inputs and outputs of a web service – the Trello web server – obtainable over a network, such as the Internet or a corporate intranet.

207. The names stored in the computer memory are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects.

Trello's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

208. The Trello computer memory also includes the address of the Trello web server, a web service.

209. Trello also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

210. The authoring tool then stores the symbolic code with the associated settings for the user-defined object in a database. It retrieves that information to produce an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player code and Application code collaborate to provide the page to the user.

211. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed

in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

212. Atlassian was made aware of the '044 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '044 patent.

213. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '044 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of Trello constitutes direct infringement of at least one claim of the '044 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising Trello as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '044 patent and knowledge that its acts were inducing infringement of the '044 patent since at least the date Atlassian received notice that such activities infringed the '044 patent.

214. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of Trello. For example, Atlassian ran an advertisement showing that users can collaborate with others to organize cards "in a fun, flexible, and rewarding way."

215. Another advertisement showed Trello's visual interface for creating, editing, and moving cards in a visual manner.

216. Since February 6, 2020, Atlassian's infringement of the '044 patent has been willful.

217. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

COUNT XII – INFRINGEMENT OF U.S. PATENT NO. 9,928,044
(JIRA)

218. The allegations set forth in the foregoing paragraphs 1 through 217 are incorporated into this Twelfth Claim for Relief.

219. Atlassian has manufactured, used, offered for sale, or sold browser-based website building tools that infringe, either literally or under the doctrine of equivalents, one or more claims of the '044 patent in violation of 35 U.S.C. § 271(a).

220. Upon information and belief, Atlassian has infringed and continues to infringe at least claim 15 of the '044 patent. JIRA practices a method for displaying content on a display of a computer device. The content is provided to the user through a web browser on webpages.

221. The method includes non-volatile computer memory that stores symbolic names associated with web components – including images, issue fields, and other web components. These components are related to inputs and outputs of a web service – the JIRA web server – obtainable over a network, such as the Internet or a corporate intranet.

222. The names stored in the computer memory are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. Each symbolic name has an associated data format class type corresponding to specific user interface objects that support that data format type, and are associated with preferred user interface objects. JIRA's code, including its HTML, JavaScript, and CSS code, associates these symbolic names – represented as element types, classes, and IDs in the browser's Document Object Model – with specific user interface objects.

223. The JIRA computer memory also includes the address of the JIRA web server, a web service.

224. JIRA also includes an authoring tool that lets users define user interface objects for presentation on the web browser. These user interface objects correspond to the web components stored in the registry. The authoring tool accesses the computer memory to select the appropriate symbolic name corresponding to the web component and associates it with the defined user interface object. A particular symbolic name is only available for particular types of user interface objects, and the defined user interface object is automatically selected by the system as the preferred object corresponding to the symbolic name of the web component selected by the user of the authoring tool. For example, a CSS stylesheet may associate all symbolic names with a given type of object.

225. The authoring tool then stores the symbolic code with the associated settings for the user-defined object in a database. It retrieves that information to produce an “Application” in the form of device-independent code, including HTML, CSS, and JavaScript code that includes the selected symbolic name. It also produces a Player in the form of device- and platform-dependent code, also including HTML, CSS, and JavaScript code. Unlike the Application code, the Player code is written for specific devices, such as laptops, tablets, or smartphones, or device platforms, such as browsers. The Player code and Application code collaborate to provide the page to the user.

226. When the Application and Player code are executed by the web browser, input provided by the user is sent to the web service, which generates output and sends it to be displayed in a user interface object on the device. The Player interprets the response and updates the user interface objects based on the output it received from the web service.

227. Atlassian was made aware of the '044 patent and its infringement thereof at least as early as February 6, 2020, when Express Mobile provided notice to Atlassian of the '044 patent.

228. Upon information and belief, since at least the time Atlassian received notice, Atlassian has induced and continues to induce others to infringe at least claim 15 of the '044 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Atlassian's partners, clients, customers, and end users, whose use of JIRA constitutes direct infringement of at least one claim of the '044 patent. In particular, Atlassian's actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising JIRA as a way for employees to collaborate on pages available on the Web. On information and belief, Atlassian has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Atlassian has had actual knowledge of the '044 patent and knowledge that its acts were inducing infringement of the '044 patent since at least the date Atlassian received notice that such activities infringed the '044 patent.

229. By way of example, Atlassian ran graphic advertisements on its website featuring infringing features of JIRA. For example, Atlassian ran an advertisement showing that users can create a webpage summarizing data entered into the platform "all in a few clicks."

230. Other JIRA advertisements touted JIRA's interface allowing for the easy creation of "flexible" and "customizable" pages.

231. Since February 6, 2020, Atlassian's infringement of the '044 patent has been willful.

232. Atlassian's infringement has damaged and continues to damage and injure Express Mobile.

JURY DEMAND

Pursuant to Rule 38 of the Federal Rules of Civil Procedure, Express Mobile demands a trial by jury on all issues triable as such.

PRAYER FOR RELIEF

WHEREFORE, Express Mobile demands judgment for itself and against Atlassian as follows:

- A. An adjudication that Defendant has infringed the '397, '755, '287, and '044 patents;
- B. An award of damages to be paid by Atlassian adequate to compensate Express Mobile for Atlassian's past infringement of the '397, '755, '287, and '044 patents, and any continuing or future infringement through the date such judgment is entered, including interest, costs, expenses and an accounting of all infringing acts including, but not limited to, those acts not presented at trial;
- C. An award of a reasonable ongoing royalty for future infringement of the '755, '287, and '044 patents;
- D. A declaration that this case is exceptional under 35 U.S.C. § 285, and an award of Express Mobile's reasonable attorneys' fees; and
- E. An award to Express Mobile of such further relief at law or in equity as the Court deems just and proper.

Dated: September 1, 2020

Respectfully submitted,

/s/ Robert Christopher Bunt

Steven F. Molo (*pro hac vice* to be filed)
NY Bar No. 4221743
Ben Quarmby (*pro hac vice* to be filed)
NY Bar No. 4298725
Leonid Grinberg (*pro hac vice* to be filed)
NY Bar No. 5652458
MOOLAMKEN LLP
430 Park Avenue, Floor 6
New York, NY 10022
(212) 607-8160 (telephone)
(212) 607-8161 (fax)
smolo@mololamken.com
bquarmby@mololamken.com
lgrinberg@mololamken.com

Robert Christopher Bunt
State Bar No. 00787165
Charles Ainsworth
State Bar No. 00783521
PARKER, BUNT & AINSWORTH, P.C.
100 E. Ferguson, Suite 418
Tyler, TX 75702
903/531-3535
E-mail: rcbunt@pbatyler.com
E-mail: charley@pbatyler.com

Rayiner I. Hashem (*pro hac vice* to be filed)
D.C. Bar No. 1046830
Sarah J. Newman (*pro hac vice* to be filed)
D.C. Bar No. 1047210
Benjamin T. Sirolli (*pro hac vice* to be filed)
D.C. Bar No. 1022426
MOOLAMKEN LLP
The Watergate, Suite 500
600 New Hampshire Avenue, N.W.
Washington, D.C. 20037
(202) 556-2000 (telephone)
(202) 556-2001 (fax)
rhashem@mololamken.com
snewman@mololamken.com
btsirolli@mololamken.com

Timothy Devlin
Delaware Bar No. 4241
DEVLIN LAW FIRM LLC
1306 N. Broom Street, 1st Floor
Wilmington, DE 19806
(302) 449-9010 (telephone)
(302) 353-4251 (fax)
tdevlin@devlinlawfirm.com

Attorneys for Plaintiff Express Mobile, Inc.